

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application. The following listing provides the amended claims with deleted material crossed out and new material underlined to show the changes made.

1. (Currently Amended) A method of logging events for a plurality of applications that operate on an operating system of a first computer, the method comprising:

~~for an event to be logged that has not yet been logged within an application:~~

receiving a request from a particular application of said applications to create an event object associated with an event of the application that has not been logged;

in response to said request, creating an the event object, ~~said event object occupying~~ in a first memory space that is ~~independent of~~ separate from a second memory space in which said application operates;

from the application, receiving temporal data and other data regarding the event;

storing ~~logging within said event object a start time, end time, the received data and~~ information regarding the event in the event object; and

analyzing ~~at least one of said start time, end time, processing the stored data and~~ information regarding the event in order to produce processed event data for display on a web browser of a second computer that is connected to the first computer through the Internet,

wherein said ~~receiving the request, said creating, said receiving the data, said logging~~ storing, and said ~~analyzing~~ processing are performed by an event logging mechanism that is part of said operating system and ~~running~~ operating independently from said application on a single the first computer on which said application ~~executes~~ operates.

2. (Previously Presented) A method according to claim 1 further comprising:

checking whether event logging has been turned on for the event.

3. (Currently Amended) A method according to claim 2, wherein said creating, receiving the data, and storing and ~~said logging~~ are performed for each event having event logging turned on, wherein a plurality of event objects are created ~~and logged~~ for a plurality of events.

4. (Currently Amended) A method according to claim 3, wherein said ~~reviewing~~ processing comprises analyzing said event object after event logging is turned off.

5. (Currently Amended) A method according to claim 4, wherein ~~analyzing~~ processing includes:

allowing user definition of the hierarchical levels of granularity of said events whose event objects are to be analyzed; and

allowing user definition of contexts for differentiating repeated occurrences of events deemed identical by nature of their hierarchical position.

6. (Currently Amended) A method according to claim 5, wherein ~~analyzing~~ processing further includes:

grouping events into their hierarchical subgroups; and

grouping events by their context, if any are defined.

7. (Currently Amended) A method according to claim 6, wherein ~~analyzing~~ processing further comprises:

traversing through the hierarchy of subgroups until the subgroup of finest granularity is traversed;

subdividing said events into further subgroups;

computing statistics for each subgroup while traversing; and

displaying said statistics.

8. (Previously Presented) A method according to claim 7, wherein if said subgroup of finest granularity has been traversed, then:

aggregating events deemed identical by virtue of their hierarchical position into an aggregate;

computing statistics for each aggregate; and

displaying said statistics for each said aggregate.

9. (Currently Amended) A method according to claim 7, wherein said ~~analyzing~~ processing-includes:

aggregating events deemed identical by virtue of their context into an aggregate;

computing statistics for each aggregate; and

displaying said statistics for each said aggregate.

10. (Currently Amended) A first computer comprising computer readable storage for storing:

~~a foundational layer upon which applications are built or executed; and~~

~~an operating system comprising an event logger; logging mechanism created by said~~ foundational layer,

~~a plurality of applications operating on said operating system,~~

~~said event logging mechanism logger executing operating independently of said~~ applications,

~~said event logger mechanism for receiving input through a web browser of a second computer to enable event logging, wherein said second computer is connected to said first computer through the Internet,~~

~~wherein when said event logging is enabled, the event logger (i) identifies identifying~~ a set of events for ~~an a particular application in the plurality of applications operating executing on~~ said ~~foundational layer operating system,~~ (ii) ~~generates generating an event log for the application,~~ and (iii) ~~records event data regarding the identified set of events in analyzing the event log, said~~

event log generated and said event data recorded in the event log without referencing any event logs of said application, ~~wherein each of said events is designated an enabled/disabled status, wherein a disabled status disables all logging for an event,~~

wherein said event logging mechanism performs the receiving, identifying, generating, and recording ~~analyzing~~ on said first computer on which said particular application ~~executes~~ operates.

11. (Currently Amended) A computer according to claim 10, wherein said event logger ~~records event data by recording logging mechanism logs~~ start time, end time, and other event ~~information data~~ into an event object for each event to be logged.

12. (Currently Amended) A computer according to claim 10, wherein said operating system ~~comprises a foundational layer is an operating system,~~ wherein said event logger operates on said foundational layer.

13. (Currently Amended) A computer according to claim ~~10~~12, wherein said foundational layer is a programmable framework.

14. (Currently Amended) A computer according to claim 10, wherein said event ~~logging mechanism~~ logger can be enabled ~~turned on~~ and then ~~off disabled from~~ beyond an execution space of said particular application operating in said operating system ~~applications within said foundational layer,~~ wherein each event in the set of events can be enabled or disabled separately ~~said turning on and off separate for each event.~~

15. (Currently Amended) A computer according to claim 10, wherein when said event logging is disabled through the ~~mechanism can be turned on and turned off and configured using a~~ web browser ~~application~~ of said second computer, said event logger foregoing the identifying, generating, and recording on said first computer on which said application operates.

16. (Currently Amended) A computer according to claim ~~15~~10, wherein said event ~~logging mechanism~~ logger is further for generating ~~generates~~ a plurality of event objects and ~~is configured to analyzing~~ analyze said event objects in order to generate an analyzed result for display on and present to said web browser application the results thereof.

17. (Currently Amended) A computer according to claim 16, wherein said event ~~logging mechanism~~ logger is configured to analyze said event objects based upon hierarchical and contextual grouping.

18. (Currently Amended) A computer to claim 16, wherein said event ~~logging mechanism~~ logger is configured to aggregate said event objects deemed identical based upon at least one of hierarchical and contextual grouping.

19. (Currently Amended) An article comprising a computer readable medium storing a computer program for execution by at least one processor, the computer program comprising a set of instructions which when executed causes:

for each event in a plurality of events ~~to be logged that has not yet been logged within relating to an application~~ (i) that generates web pages and responds to selections received through said generated web pages and (ii) that operates on an operating system of a computer,

receiving creating an event data related to at least one of a request for a web page, generation of a web page, and a request received through a selection of an item of the generated web page ~~object, said event object occupying a memory space that is independent of said application; and~~

recording logging within said event data object a start time, end time, and information regarding the event; and analyzing at least one of said start time, end time, and information regarding the event,

wherein said ~~creating receiving, logging, analyzing and recording~~ are performed by an event logging mechanism that is part of the operating system of said computer and that runs ~~running~~ independently from said application on ~~a single~~ said computer on which said application executes.

20. (Currently Amended) An article according to claim 19, wherein the ~~computer program further~~ comprises a framework, said framework comprising the event logging mechanism set of instructions which when executed causes: analyzing of said event objects according to hierarchical and contextual grouping.

21. (Canceled)

22. (Canceled)

23. (Currently Amended) A computer comprising computer readable storage for storing:
an operating system comprising foundational layer upon which applications are executed; an event logging mechanism event logger for execution on in said foundational layer, said mechanism executing independently of said;

a plurality of applications for operating on said operating system; said mechanism

the event logger for ~~for~~ (i) recording event data for each of a plurality identifying a set of events of a plurality of for an applications in a first memory space that is uniquely allocated for the event logger and is separate from a second memory space allocated for the plurality of applications; executing on said foundational layer, and (ii) grouping a plurality of said event data into a plurality of hierarchical sets of events, wherein each hierarchical set comprises a parent event and at least one child event; and (iii) for a plurality of said hierarchical sets of events, defining a nested hierarchical display of event data that displays the parent events at one aligned level in the hierarchy and displays the child events at another aligned level in the hierarchy under the corresponding parent events

~~generating a hierarchical event log for display in a web browser, said event log generated without referencing any event logs of said application and comprising an event in the hierarchy that includes a sub-event, wherein said event logging mechanism performs the identifying and generating on said computer on which said application executes.~~

24. (Currently Amended) A computer according to claim 23, wherein ~~said generating an event log~~ operating system further comprises a framework, said framework comprising said event logger ~~storing, for each event to be logged, a temporal attribute of an event in an event object associated with the event.~~

25. (Currently Amended) A computer according to claim 23, wherein ~~said event logging mechanism~~ event logger is further for analyzing said event log according to hierarchical and contextual grouping.

26. (Currently Amended) A computer according to claim 23 further comprising a first area of memory allocated to ~~the~~ a particular application of said applications, a second area of memory allocated to the ~~event logging mechanism~~ event logger, wherein said first area of memory allocated to the particular application is separate from the second area allocated to the ~~event logging mechanism~~ event logger.

27. (Currently Amended) A computer according to claim 23, wherein ~~said event logging mechanism~~ event logger can be turned on, turned off, and configured using ~~the~~ a web browser.

28. (Currently Amended) A computer according to claim 23, wherein ~~said event logging mechanism~~ event logger is further for allowing a user to enable and disable event logging for each event in the ~~set~~ plurality of events, wherein ~~said generating the recording event log data~~ is performed for each event having event logging enabled.

29. (Currently Amended) A computer according to claim 23, wherein ~~the foundational layer~~ framework ~~is an operating system~~ a framework upon which applications are executed.

30. (Currently Amended) An event logging method comprising:

for each of a plurality of events that need to be logged ~~but have not yet been logged~~
within a plurality of applications operating on an operating system of a computer:

~~creating an~~recording event-object data for each of a plurality of application
~~events; storing said event object~~ in a first memory space that is uniquely allocated for the event
logging method, said first memory space separate from a second memory space allocated for the
plurality of applications;

~~logging within said event object a start time, end time, and information~~
~~regarding the event; and~~

~~analyzing at least one of said start time, end time, and information regarding~~
~~the event,~~

for each application in said plurality of application, grouping a plurality of said event
data into a plurality of hierarchical sets of events, wherein each hierarchical set comprises a parent
event and at least one child event; and

for a plurality of said hierarchical sets of events, defining a nested hierarchical
display of event data that displays the parent events at one aligned level in the hierarchy and displays
the child events at another aligned level in the hierarchy under the corresponding parent events,
wherein ~~said creating recording, storing, logging, and analyzing grouping, and defining~~ are
performed by an event logging mechanism that is part of the operating system of said computer and
that runs ~~running~~ independently from said applications on ~~a single~~ said computer.

31. (Currently Amended) A method according to claim 30 ~~further comprising creating,~~
~~for the event, an enabled/disabled status,~~ wherein the operating system comprises a framework, said
framework comprising disabled status disables all logging for the event logging mechanism within a
system that includes the plurality of applications.

32. (Currently Amended) A method according to claim 30 further comprising checking, for each event identified by a ~~[[n]]~~ particular application within the plurality of applications, whether event logging has been enabled.

33. (Currently Amended) A method according to claim 32, wherein said nested hierarchical display analysis is displayed in a web browser ~~performed after event logging is disabled~~.

34. (Currently Amended) A method according to claim 30, wherein the memory space occupied by the event ~~log~~ data is within a memory space that has been allocated solely to the event logging mechanism.

35. (Previously Presented) A method according to claim 30, wherein the events that are logged by the event logging mechanism have not been previously logged by any other application.

36. (Currently Amended) A method according to claim 30, wherein information placed in the event ~~log~~ data is first logged by the event logging mechanism.

37. (Currently Amended) A method according to claim 30, ~~wherein further comprising~~ each event comprises an enable/disable state ~~for each event~~, wherein ~~the~~ a disable state of a particular event precludes any system from logging said particular ~~creating an event log~~.

38. (Currently Amended) A method according to claim ~~30~~ 33, wherein said computer is a first computer, wherein said web browser executes on a second computer that is connected to said first computer through the Internet ~~creating is done by a foundational layer that is a development framework~~.

39. (Currently Amended) A method of logging events for an application, said method comprising:

receiving ~~identifying~~ a set of events generated by said application; ~~and~~

creating a corresponding set of event data in a first memory space that is separate from a second memory space in which the application executes;

analyzing the set of event data by determining a procedural level at which each event is executed in the application;

hierarchically ~~logging~~ grouping said ~~identified~~ set of event[[s]] data into a plurality of groups based on said analysis of the set of event data, wherein each event executed at a same procedural level in the application is grouped in a same group; ~~wherein at least one event in the hierarchy comprises a sub-event;~~

defining a hierarchical display of event data comprising a nested hierarchical presentation of data for each of said events based on said grouping, said hierarchical display presented in a web browser, wherein said ~~identifying~~ receiving, creating, analyzing, grouping, and said ~~logging~~ generating are performed by an event logging mechanism running independently from said application on a single computer on which said application executes.

40. (Currently Amended) A method according to claim 39, wherein said web browser is a web browser executing on said single computer ~~further comprising:~~

~~checking whether event logging has been turned on for the event;~~

~~wherein said creating and said logging are performed for each event having event logging turned on, wherein a plurality of event objects are created and logged for a plurality of events.~~

41. (Currently Amended) A method according to claim ~~40~~ 39, wherein said single computer is a first computer, wherein said web browser executes on a second computer that is connected to said first computer through the Internet further comprising:

~~analyzing said event objects after event logging is turned off.~~

42. (Currently Amended) A computer comprising computer readable storage for storing:
an operating system comprising an ~~foundational layer upon which applications are~~
~~executed~~ event logger; and

a plurality of applications operating on said operating system, wherein at least one particular application is for generating web pages and responding to web page selections received through said generated web pages,

wherein the event logger~~an event logging mechanism~~ is for:

(i) execution on said foundational layer,~~for~~ functioning interoperably with but separately from said applications, ~~said mechanism for:~~

(ii) identifying a set of plurality of event data for a plurality of application events of the particular an application-executing on said foundational layer, said application events comprising a request for a web page, generation of a web page, and a request received through a selection of an item of the generated web page

(iii) storing~~generating an~~ the event data in an application event log to record
~~said identified event data,~~

~~analyzing said event data, wherein said application does not generate an event log.~~

43. (Currently Amended) A computer according to claim 42, wherein said storing said event data~~generating an event log~~ comprises storing, for each event to be logged, a start time, end time, and information regarding the event.

44. (Currently Amended) A computer according to claim 42, wherein said ~~event logging mechanism comprises~~ event logger is further for analyzing of said event log according to hierarchical and contextual grouping.

45. (Currently Amended) A computer according to claim 42[[44]], wherein said plurality of application events comprises ~~further comprising an enable/disable state for each event identified by the application,~~ wherein the disable state precludes the ~~system~~ the event logger from creating an

event log, wherein the identifying and storing are ~~generating an event log is~~ performed for each event having event logging enabled.

46. (Previously Presented) A method comprising:

- identifying a set of events that has not been logged by an application;
- analyzing the set of events identified for the application, wherein each event comprises at least a start time and an end time;
- grouping the set of events based on said analysis of the set of events; and
- generating a display of said set of events based on said grouping, wherein said identifying, analyzing, grouping, and generating are performed by an event analysis mechanism running on a single computer on which said application runs, said event analysis mechanism running independently from said application on said computer.

47. (Previously Presented) The method of claim 46, wherein said set of events are identified by an event logging mechanism prior to performing said analyzing, grouping, and generating.

48. (Previously Presented) The method of claim 46, wherein said generated display is for displaying said groupings of said set of events in a web browser.

49. (Previously Presented) The method of claim 46, wherein said grouping comprises grouping said set of events into a hierarchy, wherein at least one event in the hierarchy includes a sub-event.

50. (Previously Presented) The method of claim 46 further comprising computing statistics using the start time and the end time of the set of events, wherein said generating comprises including the statistics in the display.